REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
2002	Conference Paper	2000 to 2002

4. TITLE AND SUBTITLE
A Feasible Approach for Implementing Greater Levels of Satellite Autonomy

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62601F

6. AUTHOR(S)
Steve Lindsay, Paul Zetocha

5d. PROJECT NUMBER
8809

5e. TASK NUMBER
LJ

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/VS
3550 Aberdeen Ave. SE
Kirtland ARB, NM 87117-5776

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; Distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In this paper, we propose a means for achieving increasingly autonomous satellite operations. We begin with a brief discussion of the current sate-of-the-art in satellite ground operations and flight software, as well as the real and perceived technical and political obstacles to increasing the levels of autonomy on today's satellites. We then present a list of system requirements that address these hindrances and include the artificial intelligence (AI) technologies with the potential to satisfy these requirements.

We conclude with a discussion of how the space industry can use this information to incorporate increased autonomy. From past experience we know that autonomy will not just "happen," and we know that the expensive course of manually intensive operations simply cannot continue. Our goal is to present the aerospace industry with an analysis that will begin moving us in the direction of autonomous operations.

15. SUBJECT TERMS
flight software, autonomy, satellites, artificial intelligence

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Paul Zetocha
Unclassified	Unclassified	Unclassified	Unlimited	7	**19b. TELEPHONE NUMBER** *(include area code)* (505) 853-4114

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

086

98-5177

A Feasible Approach for Implementing Greater Levels of Satellite Autonomy

Steve Lindsay
Air Force Research Laboratory/VSSS
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776
(505) 853-4140, fax: (505) 846-6053
steve.lindsay@vs.afrl.af.mil

Paul Zetocha
Air Force Research Laboratory/VSSS
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776
(505) 853-4114, fax: (505) 846-6053
paul.zetocha@vs.afrl.af.mil

ABSTRACT

In this paper, we propose a means for achieving increasingly autonomous satellite operations. We begin with a brief discussion of the current state-of-the-art in satellite ground operations and flight software, as well as the real and perceived technical and political obstacles to increasing the levels of autonomy on today's satellites. We then present a list of system requirements that address these hindrances and include the artificial intelligence (AI) technologies with the potential to satisfy these requirements.

We conclude with a discussion of how the space industry can use this information to incorporate increased autonomy. From past experience we know that autonomy will not just "happen," and we know that the expensive course of manually intensive operations simply cannot continue. Our goal is to present the aerospace industry with an analysis that will begin moving us in the direction of autonomous operations.

1. INTRODUCTION

The operations of an average satellite ground station are manually intensive. In a typical Air Force Satellite Operation Center (SOC) very little automation exists in either the ground station or on-board the space vehicle (SV). With the exception of some of the newer ground stations, downlinked telemetry is displayed in lengthy textual format with very little automated capability to decipher and diagnose the health and status (H&S) of a SV. Detecting, diagnosing, and troubleshooting problems is almost entirely dependent on the expertise of the operator and/or satellite engineer. Commanding is generally a manual process with a human operator in the loop for command verification. Furthermore, with the lack of ground automation, the probability of human error is much higher than it needs to be. Commercial satellite ground stations generally have more extensive automation primarily because a) they are typically operating satellites which have been on-orbit for a short amount of time, and b) they have tighter budget constraints dictating a need for increased automation. Nevertheless the technology exists to further increase the automation of commercial SOCs from what it is today.

In addition, all satellites exhibit some degree of autonomy, as virtually none are monitored and commanded 24 hours per day. Even for satellites in geosynchronous orbit H&S checks are typically accomplished only a few times each day at regular intervals. In general, the degree of autonomy that a typical satellite flying today currently exhibits is largely a function of necessity. For instance, attitude control is an autonomous function because it must be. Performing attitude determination and control (ADAC) on the ground would be disastrous. On the other hand orbit determination and control is almost entirely done on the ground, as this function is generally done only on a predetermined basis with the time, within coarse limits, known well in advance. To reduce perceived risk ADAC functions are done on the ground where human control and oversight is possible.

In the AFRL Satellite Autonomy Program, we define "satellite autonomy" as a combination of ground automation and on board autonomy. We use this definition because of the fact that all satellite operations functions are performed in at least one of three locations: manually on the ground, automated on the ground, or autonomously on board. Because the space industry's number one concern is to mitigate risk to their multimillion-dollar space-borne assets and terrestrial subscribers, any attempts to further the cause of autonomous operations must be proven successful in all three locations, in turn. Clearly, the space industry doesn't want to take any risks they deem unnecessary, and for good reason; they have a long history of successes and they wish to continue this trend. In fact, some companies who have entire Internal Research and Development programs devoted to achieving higher levels

of autonomy are continually having to justify their existence time and again to upper management.

In the next two sections of this paper we will investigate some of the impediments to enhancing the level of satellite autonomy and look at how some of these impediments might be addressed and ultimately removed.

2. IMPEDIMENTS TO ACHIEVING NEW LEVELS OF AUTONOMY

Many people in the aerospace industry perceive autonomy as an evil rather than a virtue. In fact, the strongest proponents are the research computer scientists who want to bring the state-of-the-art to the operations world, while the most adamant opponents are the program managers and satellite operators themselves. Clearly, before rallying support in favor of autonomy, we need to develop a firm understanding of why the subject tends to meet such strong resistance.

Table 1 lists the common arguments against satellite autonomy. At the top of the list is the notion that a satellite cannot be trusted to make its own complex decisions. In reality, today's operational satellites continuously make their own decisions, as no satellite is commanded around the clock. For example, each satellite in the DSCS-III geosynchronous constellation incorporates closed-loop attitude control by receiving input from its sun and earth sensors and inertial reference unit, and commanding its reaction wheels and thrusters, every 16 seconds. However, this and the approximate eight other autonomous functions on DSCS-III is implemented with conventional software in a control loop, offering minimal flexibility in implementing additional autonomous functions and enormous inflexibility in modifying the ones that currently exist.

The second impediment is that operators feel an autonomously orbiting satellite translates to an asset for which they are responsible but over which they have no control. A common operator viewpoint is that the flight software engineers programmed the control directly into the flight software, leaving the operators little say as to how it executes. This viewpoint represents a common misperception that on board intelligence must be implemented "all or nothing." Of course, such an approach would be doomed from the outset, as the first software glitch would cause the entire aerospace industry to identify autonomy as an evil to be avoided at all costs. In implementing autonomy a cost-benefit tradeoff analysis must be done and choices have to be made as to what functionality should be made autonomous. To many people artificial intelligence (AI) has a bad connotation; they perceive all AI software as entirely non-deterministic with little opportunity to incorporate failsafe modes.

Furthermore, critics say the up-front software development costs are excessive when compared to the relatively minimal costs of staffing the ground center with a few extra people. If extensive custom software development were required to implement greater levels of autonomy then this would certainly be the case. However, COTS products exist today that provide the control infrastructure for implementing autonomy (the "how"), while allowing the software development team to focus on the autonomous behavior and interaction of the individual components (the "what"). Not only is this type of development environment inexpensive, it also allows members of the operations and engineering team to develop software themselves, rather than passing requirements to a team of computer scientists who may know little about the flight environment and who certainly represent an additional chance for miscommunication. Many of the AI COTS tools available were designed to be extensible allowing autonomy to be implemented in a gradual way with more and more functionality added as operators become comfortable with the software's decision making capability.

The fourth impediment is that autonomous flight software is perceived difficult to verify and validate (V&V), as opponents contend the flight environment can never be adequately simulated on the ground. High-fidelity simulation software will certainly help in this respect, especially when combined with fully deterministic software. It should be noted that this argument is not specific to autonomous flight software but really applies to software in general. In any complex system it becomes extremely difficult to envision *a priori* all possible scenarios which may arise. An additional advantage could be gained by striving for "certification," as opposed to validation. For example, consider the way a craftsman trains his apprentice. He first allows the apprentice to observe, then supervises the apprentice in action, and eventually lets his student operate without supervision. If an operator were able to "train" flight software in this manner, a satellite would more easily achieve its goal of autonomous operation.[4]

The next three impediments listed in Table 1 are political, rather than technological, in nature. Satellite hardware manufacturers and software developers are understandably reluctant to release design information to software companies for developing tailored AI tools. However, nondisclosure agreements and contractual incentives supporting collaboration can be used to alleviate, though not totally eliminate, these concerns. In general, there has to be some benefit to be gained by an organization before information is turned over for use by another organization. Impediment six is certainly understandable, as operations personnel may feel their jobs are in jeopardy by automating a satellite system. Furthermore, once an automated system is fielded, these operators perceive their jobs as less prestigious because less skill is required. Impediment seven is undoubtedly the most difficult to address, as it requires a change to how ground station supervisory positions are perceived. Fortunately, these three impediments will inevitably

less skill is required. Impediment seven is undoubtedly the most difficult to address, as it requires a change to how ground station supervisory positions are perceived. Fortunately, these three impediments will inevitably disappear, as budgets continue to shrink and ground automation becomes more a necessity than a luxury.

The next impediment is an argument we often hear: it is difficult to quantify the benefits of autonomy. More quantitative studies are needed which provide accurate cost savings to be gained by making certain functionality more autonomous. We also need to determine what functionality should be made autonomous and what should remain within operator control. Until the benefits of autonomy can be accurately quantified and the cost savings made apparent, satellite program offices will be reluctant to buy into the concept.

The last impediment concerns satellites that are already on orbit. For satellites with older architectures it may not be possible to do much enhancement of the on board flight software. Ground stations can be upgraded or replaced entirely. The impediment to doing this, however, is the much used "don't fix it if it ain't broke" argument. This is a valid argument and the cost of upgrading a ground station must be weighed against the future cost savings that will be achieved by doing so.

No.	Description
1	It's too risky to let an orbiting satellite make its own complex decisions
2	Operators/engineers fear an inability to control the actions of an autonomous satellite
3	Autonomous software development seems too costly when compared to ops personnel
4	Autonomous flight software is difficult to verify and validate
5	Satellite manufacturers are reluctant to release proprietary info to AI software developers
6	Ground automation is evaluated by operations personnel who feel their positions will be rendered obsolete
7	Ground facility supervisors derive their prestige from the number of people, rather than the number of computers, in an operations center
8	A quantitative analysis of the benefits of autonomy has never been performed
9	Operational satellites will receive minimal benefit from autonomous ground software

Table 1. Impediments to Autonomy

3. REQUIREMENTS THAT ADDRESS THE IMPEDIMENTS TO AUTONOMY

In this section we formulate a set of software requirements that address the technological impediments listed in Table 1. Once these requirements are met, the aerospace industry will be well on its way to achieving higher levels of autonomous satellite operations. We hope to show that by incorporating recent but proven AI technology, on-orbit capability is greatly increased while risk to the vehicle is minimized.

Table 2 summarizes these requirements. The first three columns list the requirement and the corresponding impediments it addresses from Table 1. The subsequent columns list different AI technologies and the degree to which they satisfy the requirements for achieving greater autonomy. A bold check mark indicates a technology that is a good candidate for satisfying that software requirement. A light check mark indicates an AI technology that may be a viable candidate in certain circumstances, while a nonexistent check mark indicates an AI technology with little ability to satisfy that particular software requirement. Because of the limited scope of this paper we chose not to explain these technologies in detail; however, we invite the reader to review the references or numerous sources available for additional information.

The analysis seeks to provide a conservative path towards achieving satellite autonomy in a way that would be acceptable to those adverse to risk. For missions not adverse to risk a reevaluation of the table is necessary.

No.	Description	Addresses Impediments	Rule-Based	Finite State-Based	Model-Based	Case-Based	Agent-Based	Neural Networks
1	Fully deterministic computing results	1, 5, 4	✓	✓	✓	✓		✓
2	Time-driven and event-driven control	1, 2	✓	✓	✓		✓	✓
3	Well leant to functional migration	2, 3, 4, 9	✓	✓	✓	✓	✓	
4	Facilitates long-term trending	1	✓	✓				✓
5	Supported by COTS products/vendors	2, 3, 4, 9	✓	✓	✓	✓	✓	✓
6	Proven track record for satellite control	1, 8	✓	✓	✓	✓		
7	Requires minimal software expertise	2, 3	✓	✓	✓	✓		

Table 2. Software Requirements Addressing the Impediments to Autonomy

The first requirement is for fully deterministic software. Due to the conservative nature of the space industry the software should not be allowed to *learn* or to be stochastically based. With this criteria finite state based systems and model-based reasoning systems are strong candidates. Finite state systems place the vehicle in known states and provide deterministic transitions between states. Non-determinism is entered into the equation if the SV happens to be in an undefined state. In that case the vehicle should be placed in a "safe" mode while ground personnel first place the satellite back in a known state and then add the new state to the system. Model-based reasoning (MBR) systems are based on satellite model component behavior and the structural relationship between these components. Given that model components are generally deterministic identical system inputs will always yield the same outputs. To a lesser extent rule-based and case-base systems and neural networks satisfy this requirement. Rule-based systems have a somewhat non-deterministic side to them in that *time* becomes a factor if multiple rules are attempting to fire at once. Rules get executed based on priority and their order on the stack. For complex large systems non-deterministic results can happen. Supervised neural networks learn based on a training data set. Non-determinism comes into play when data is entered into the system that is different from data used to train the system.

Autonomous software should incorporate both time and event driven control. This allows a spacecraft to carry out complex operations within tight operational constraints, thus minimizing risk. Finite state systems clearly provide the capability to transition between states as a function of either time or event. Rule-based systems are inherently event-driven systems; however, rules can be made to fire as a function of time and rule priorities can alter the execution of rules on the stack. Agent-based systems are implementation dependent and thus can be either time or event driven. In general we see MBR systems and neural network implementations as software utilities that are evoked by other processes and thus their semantics are dependent on the calling routine.

Earlier we stated that autonomy should not be viewed as an all or nothing choice but should be implemented gradually as the operations community becomes more comfortable with its performance. Consequently, we feel AI systems that easily allow extension to their knowledge bases are favorable. Most AI systems offer this extensibility. In general, new information regarding satellite behavior is easily handled as new rules to an expert system knowledge base, new states in a state space system, or new cases in a case library of a case-based reasoning (CBR) system. These systems can be implemented such that the additions do not have an adverse effect on the existing knowledge bases. Extensibility to an MBR system can be achieved if that capability is designed from the start.

Although not a necessity, a desirable trait of autonomous software is the support of long-term trending. Compared to a system that functions only on discrete changes in telemetry data, a system capable of detecting trends and predicting anomalous behavior is more beneficial. Within states finite state space systems have this capability and rules can be made to fire based on trends in data. Neural networks have proven successful in the past when used for trend analysis though their applicability to the satellite domain is still relatively new.

Where feasible, software should be supported by commercial-off-the-shelf (COTS) products and or vendors. COTS software is generally far less expensive than if an identical capability had to be developed from scratch. In addition COTS software will generally have undergone extensive testing before release thus partially satisfying the requirement for software verification and validation. COTS vendors support almost all of the AI technologies in

have really been proven in ground control as numerous prototypes have been implemented and are in operational use. To a lesser extent rule-based systems have made headway into flight software; however, the general user community appears to have not fully accepted this. Progress has been made in that direction over the last few years and it is becoming more acceptable. Finite-state systems are newer but are making their way into a number of ground systems. CBR and MBR systems have also been used in ground control systems, although their use has not been widespread enough to claim they have a proven track record. Numerous issues are involved with implementing MBR and CBR systems on a large scale that need to be taken into consideration, such as having access to good high fidelity models, accounting for the computational complexity of MBR systems, and having access to a large case library.

Software should also require minimal expertise to implement from a user standpoint. In that respect both rule-based systems and finite state systems are fairly easy to implement. With both of those systems the bulk of the effort is spent up front deciphering available documentation and attempting to understand the functionality of the satellite. Given appropriate MBR and CBR engines those systems can potentially be easy to implement and would involve building a model base or case library. Again the bulk of the effort would be spent up front during the knowledge acquisition phase.

4. A FEASIBLE APPROACH

Clearly two political battles must be won. First, the impediments must be addressed and resolved one by one as we attempted to do in the previous sections. This will provide developers at least the opportunity to implement new levels of autonomy. As Table 2 shows the software technologies needed to implement autonomous satellite systems currently exist. No single AI technology is capable of satisfying all requirements addressing the impediments to autonomy; rather, the ideal solution is a mixture of AI technologies in conjunction with more *traditional* software. For example, a state-space or rule-based system might be used to monitor, via telemetry data, the overall status of a vehicle and identify *known* anomalous situations. An MBR or CBR system might be called upon to handle *unknown* anomalies and a neural network might be used for the control of a specific component that exhibits nonlinear behavior. An agent-based architecture might then be used to integrate the various components on-board the satellite, on the ground, and on-board other peer satellites as well as to foster cooperation amongst these entities.

A more important battle is ensuring that implementation is exercised to its fullest potential. No matter how successful the software is in theory, if satellite operators have no motivation to exercise it—or worse, if they have opportunities to sabotage it—then autonomy will be dealt a harsh blow. This becomes a difficult issue to address if the goal is to reduce manpower and an operator feels his or her job may be in jeopardy. To be successful operators left remaining should view the software involved in the move towards autonomous operations as tools designed to alleviate them from more mundane tasks so that more time is available to be spent on tasks of greater importance.

The implementation of autonomy should be accomplished in stages and we feel an architecture that supports incremental implementation is an absolute necessity. This architecture would ideally support a knowledge base framework that would reside both on the ground and on board the satellite. The contents of the two knowledge bases would not be similar but could easily be updated during on-orbit operations. Upon initial implementation the ground systems knowledge base would handle virtually all tasks while the on-orbit system would handle only mundane tasks. To reduce perceived risk initially the ground system would be configured to have more of an advisor role rather than that of autonomous H&S maintenance and command and control. Upon detection of anything relevant the system would offer advisement regarding the appropriate course of action. As time goes on and an operator becomes more comfortable with the decision making abilities of the ground system concerning certain actions those abilities could be automated. Likewise functionality would be slowly migrated from the ground knowledge base to the on-orbit knowledge base with lower risk tasks migrated first. As before, to reduce perceived risk the on-board software would first function in an advisor role and automate incrementally as operations personnel become more comfortable with its actions. While the above serves to offer one level of software validation, extensive V&V should be done prior to on-orbit operations through ground based simulations in a realistic test environment.

Virtually all satellite program offices are reluctant to implement high levels of autonomy without the technology first being proven. These organizations generally prefer to see it implemented "on the other guys' system first." Two avenues are available to help get around this. The first is to implement autonomy as an experiment onboard any of the numerous experimental satellites flown out of our national laboratories. Incorporating failsafe modes to prevent the software from performing any catastrophic functions could reduce perceived risk to these satellites. The second option available for implementing autonomy is to use existing satellites that are still on orbit but that have completed their missions. These vehicles could potentially be used as on-orbit testbeds. A number of issues would need to be considered such as the state of the vehicle after mission life, the amount of consumables available, the reconfigurability of the vehicle in terms of uploading autonomous software, and the use of existing ground operations facilities.

5. CONCLUSIONS

The continuously increasing costs of software development and ground operations personnel, coupled with recent advances in artificial intelligence, provided our motivation for analyzing how best to approach and implement autonomy. The impediments to increasing autonomy are no longer technological—they are political. As budgets continue to shrink even these impediments will eventually disappear, but no one know how long that will take.

The best thing we in the space industry can do is to develop flight experiments demonstrating the effectiveness of autonomy. We at the Air Force Research Laboratory (AFRL) are fortunate to have the MightySat program, which is an inexpensive bus that exists solely to demonstrate technologies developed and sponsored by AFRL, and the Intelligent Satellite Systems section has numerous experiments proposed for flights beginning in January 2000. Once we and other R&D organizations begin successfully demonstrating cost-effective implementations of satellite autonomy, the space industry will be well on its way to realizing the benefits of these technologies.

6. REFERENCES

1. Zetocha, Paul; Frostman, Dave; Statsinger, Robert, "*Towards Autonomous Space Systems*", Proceedings of the Artificial Intelligence / Expert System Symposium, Los Angeles CA, 1994.
2. Zetocha, Paul; Ortiz James, "*PEACH - An Agent for Increased Space Operations Automation*", Proceedings of the SpaceOps 98 conference, Tokyo Japan, 1998.
3. Pell, Barney, Scott Sawyer, Nicola Muscettola, Ben Smith, Douglas Bernard, "*Mission Operations with an Autonomous Agent*".
4. Pell, Barney: "The Future of Autonomy," Software for Space Systems Autonomy Workshop, Albuquerque, NM, December 1997.
5. Bernard, D.E.; Dorais, G. A.; Fry, C.; Jr., E. B. G.; Kanefsky, B.; Kurien, J.; Miller, W.; Muscetolla, N.; Nayak, P. P.; Pell, B.; Rajan, K.; Rouquette, N.; Smith, B.; and Williams, B. C., *Design of the Remote Agent Experiment for Satellite Autonomy,* Proceedings of the IEEE Aerospace Conference, Snowmass CO, 1998.
6. Zetocha, Paul, *Model-Based Reasoning for Satellite Anomaly Diagnosis,* Proceedings of the 2nd Int'l Symposium on Reducing the Cost of Ground Operations, Oxford UK, July 1997.